Studia i Materiały Informatyki Stosowanej, Tom 18, Nr 1, 2026
str. 27-29

ZASTOSOWANIE WYBRANYCH ALGORYTMÓW KRYPTOGRAFICZNYCH W APLIKACJI KOMUNIKATORA END-TO-END

Dawid Birski, Piotr Kotlarz

Uniwersytet Kazimierza Wielkiego
Wydział Informatyki
ul. Mikołaja Kopernika, 85-074 Bydgoszcz
e-mail: dawid.birski@ukw.edu.pl

Streszczenie: *Niniejszy artykuł opisuje projekt oraz implementację prostego komunikatora tekstowego z szyfrowaniem hybrydowym end-to-end, działającego na platformach Windows i Android z wykorzystaniem serwera Firebase. Popularne komunikatory są często niebezpieczne ze względu na brak domyślnego szyfrowania end-to-end. Projekt obejmuje generowanie kluczy, szyfrowanie i odszyfrowywanie wiadomości, oraz międzyplatformową komunikację między wieloma użytkownikami. Projekt udowadnia, że samodzielnie można stworzyć rozwiązanie oferujące wysoki poziom bezpieczeństwa rozmów tekstowych, porównywalny z najlepszymi aplikacjami rynkowymi*.

Słowa kluczowe: *Szyfrowanie, bezpieczeństwo, kryptologia, komunikatory*

Application of selected cryptographic algorithms in an end-to-end messenger application

Abstarct: *This article presents the project and implementation of a simple text messenger with hybrid end-to-end encryption running on Windows and Android platforms using a Firebase server. Popular messengers are often unsafe due to the lack of default end-to-end encryption. The project covers key generation, message encryption and decryption, and cross-platform multi-user communication. The project demonstrates that it is possible to independently create a solution offering a high level of text conversation security, comparable to the best market applications.*

Keywords: *Encryption, security, cryptology, messengers*

1. WSTĘP

Kryptografia ma rosnące znaczenie w ochronie danych prywatnych w komunikatorach wobec cyberzagrożeń. Popularne komunikatory, powstałe w XXI wieku, początkowo nie stosowały szyfrowania end-to-end, mimo dostępności algorytmów DES i RSA od 1977 r., z AES zastępującym DES w 2001 r., umożliwiających hybrydowe szyfrowanie. Wiele z nich nadal nie oferuje szyfrowania end-to-end domyślnie lub w ogóle. Implementacja takiego szyfrowania jest istotna dla bezpieczniejszej komunikacji, bez podatności na przechwytywanie i włamanie.

2. ROZEZNANIE RYNKU

Autorski komunikator wykorzystuje szyfr symetryczny AES-256 i asymetryczny RSA-2048, z domyślnym szyfrowaniem end-to-end.

Signal od początku istnienia, WhatsApp od 2014 roku, i Messenger od 2023 roku wykorzystują szyfr symetryczny AES-256 i asymetryczny Curve25519, z domyślnym szyfrowaniem end-to-end. Są one bezpieczne na poziomie komunikatora z projektu [3][6].

Telegram od początku istnienia, Messenger od 2016 roku, TeamSpeak 3, i Skype od 2018 roku wykorzystują szyfr symetryczny AES-256, Telegram wykorzystuje szyfr asymetryczny RSA-2048, Messenger i TeamSpeak 3 Curve25519, a Skype RSA-1536 lub RSA-2048, jednak

wszystkie te komunikatory mają jedynie opcjonalne szyfrowanie end-to-end, przez co są częściowo podatne na włamania [4, 7, 9-10, 13-14]

Skype od 2003 do 2018 roku wykorzystywał szyfr symetryczny AES-256 i asymetryczny RSA-1536 lub RSA-2048, bez żadnej opcji szyfrowania end-to-end [11].

Gmail od 2010 roku, Messenger od 2011 do 2023 roku, Discord od początku istnienia, wykorzystywały bądź dalej wykorzystują protokół TLS do szyfrowania symetrycznego i asymetrycznego, bez żadnej opcji szyfrowania end-to-end, przez co są w pełni podatne na włamania [5, 8, 12].

Gmail od 2004 do 2010 roku wykorzystywał protokół TLS do szyfrowania jedynie opcjonalnie, bez żadnej opcji szyfrowania end-to-end, przez co był podatny na przechwytywanie wiadomości, jak i na włamania [12]

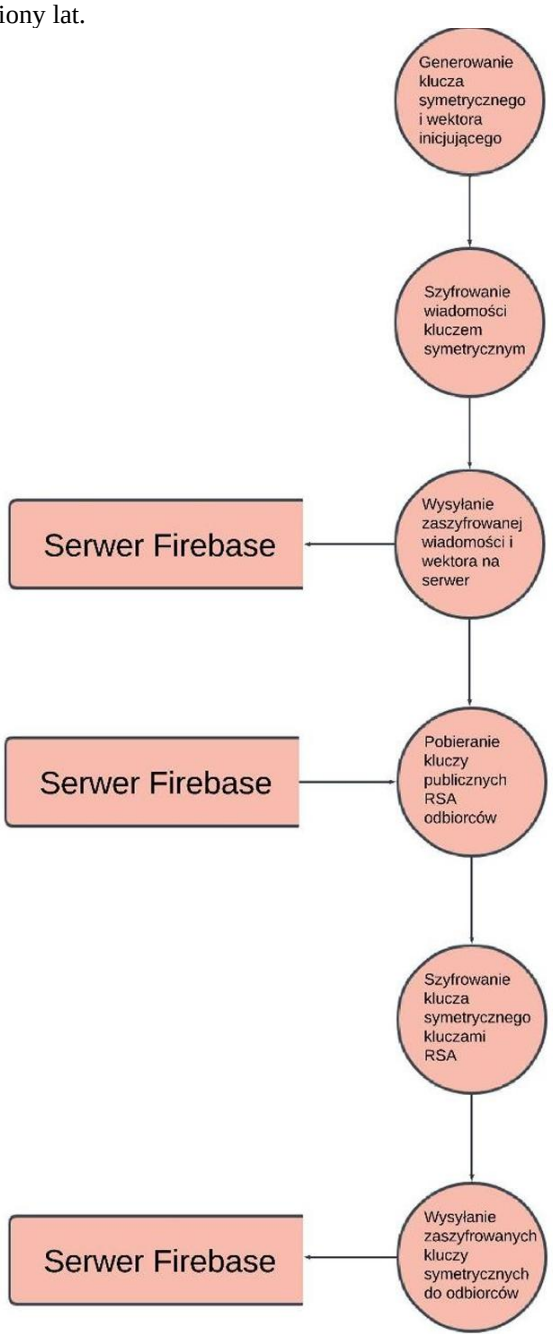
WhatsApp od 2012 do 2014 roku wykorzystywał jedynie niebezpieczny szyfr symetryczny RC4 i żadnego szyfru asymetrycznego, bez żadnej opcji szyfrowania end-to-end, przez co był praktycznie niezabezpieczony [2].

WhatsApp od 2009 do 2012 roku nie wykorzystywał jakichkolwiek szyfrów, i był kompletnie niezabezpieczony [1].

3. OPIS PROJEKTU

Autorski komunikator wykorzystuje szyfrowanie hybrydowe (rys. 1), czyli takie, które łączy szyfrowanie symetryczne z asymetrycznym, oraz metodę szyfrowania end-to-end, czyli pomiędzy nadawcą i odbiorcą, w taki sposób, że tylko urządzenie nadawcy, który sporządza wiadomość, i odbiorca, który czyta wiadomość, ma dostęp do odszyfrowanej wiadomości. Dzięki temu, bez dostępu do maszyny nadawcy bądź odbiorcy, dane są niemożliwe do przechwycenia, i nie wyciekną w wypadku włamania na serwer komunikatora, który przesyła wiadomości między użytkownikami.

Szyfrowanie symetryczne jest bardziej wydajne, dlatego jest używane do zaszyfrowania samej wiadomości zamiast szyfru asymetrycznego. Jednak szyfrowanie asymetryczne jest niezbędne, ponieważ tylko ono jest nieprzechwytywalne, i jest używane do zaszyfrowania klucza symetrycznego. Szyfrowanie asymetryczne działa w taki sposób, że zamiast jednego klucza są dwa – jeden do zaszyfrowania wiadomości, a drugi do odszyfrowania. Odbiorca generuje oba klucze w momencie logowania do aplikacji, zachowując ten do odszyfrowania, to jest klucz prywatny, a do nadawcy wysyła klucz publiczny, którym on szyfruje klucz symetryczny po swojej stronie, przesyłając go do odbiorcy wraz z zaszyfrowaną wiadomością. Haker podsłuchujący transfer danych ma więc dostęp do zaszyfrowanej wiadomości i zaszyfrowanego klucza symetrycznego, ale nie może ich odczytać ani wykorzystać. Bowiem dzisiejsze szyfry są tak silne, że nawet z mocą obliczeniową całego świata złamanie jednego zajęłoby miliony lat.



Rysunek 1 Uproszczony schemat wysyłania wiadomości

4. PODSUMOWANIE I WNIOSKI

W niniejszym artykule przedstawiono samodzielnie wykonany komunikator tekstowy z szyfrowaniem hybrydowym end-to-end, na poziomie najbezpieczniejszych na rynku aplikacji, co świadczy o tym, że stosunkowo prosto jest taki projekt wykonać, a mimo to większość komunikatorów należących do całych korporacji nie posiada tych funkcji. Najprawdopodobniej to jest celowe, gdyż jeśli te korporacje chciałyby zaimplementować te zabezpieczenia, miałyby do tego zdolność. Jednak rezygnacja z szyfrowania end-to-end umożliwia przechowywanie odszyfrowanych wiadomości na serwerach korporacji, co może być używane do analizy i sprzedaży danych. Dla zainteresowanych autor udostępni aplikację i dalsze informacje drogą mailową.

Literatura

1. https://en.wikipedia.org/wiki/Reception_and_criticism_of_WhatsApp_security_and_privacy_features
2. https://www.computerworld.com/article/1524242/whatsapp-flaw-could-allow-attackers-to-decrypt-messages.html
3. https://www.bbc.com/news/technology-30114346
4. https://membit.io/blog/is-telegram-encrypted-what-that-means-for-corporate-communication/
5. https://www.purevpn.com/encrypted-tunnel/discord-encryption
6. https://about.fb.com/news/2023/12/default-end-to-end-encryption-on-messenger/
7. https://about.fb.com/news/2016/07/messenger-starts-testing-end-to-end-encryption-with-secret-conversations/
8. https://www.helpnetsecurity.com/2012/11/19/facebook-rolls-out-always-on-https-by-default/
9. https://support.microsoft.com/en-us/skype/what-are-skype-private-conversations-f64639c9-a156-4928-97fe-753ac0c6a943
10. https://support.microsoft.com/en-us/skype/does-skype-use-encryption-28f5b30b-fcce-493a-9e55-049add6c2d39
11. https://www.comparitech.com/blog/information-security/is-skype-safe-and-secure-what-are-the-alternatives/
12. https://support.google.com/transparencyreport/answer/7381230?hl=en#zippy=%2Chow-does-encryption-in-transit-relate-to-https-access-to-gmail
13. https://github.com/ReSpeak/tsdeclarations/blob/master/ts3protocol.md
14. https://support.teamspeak.com/hc/en-us/articles/4523999467933-Rooms-How-do-I-join-a-room-how-do-I-create-a-room
15. https://support.teamspeak.com/hc/en-us/articles/360002711378-Does-TeamSpeak-3-encrypt-my-voice-packets
16. https://pl.wikipedia.org/wiki/Kryptologia
17. https://www.sectigo.com/resource-library/rsa-vs-dsa-vs-ecc-encryption